It's Not Easy Building Green: Predicting Lead time for LEED Certifications

BUDT733

Team #2
Lauren DeStefano, Anu Gupta, Tim Lewis and Scott Lewis

GREEN BUILDING™
CERTIFICATION INSTITUTE

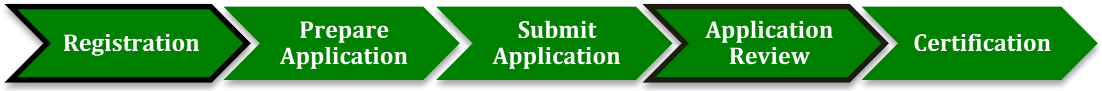
Executive Summary

Background

The Green Building Certification Institute (GBCI) is an independent, third party organization committed to promoting the design, development, and implementation of processes used to increase and measure green building performance and green building practice. GBCI provides two main services to its customers. First, GBCI is qualified to bestow onto individuals LEED Green Associate and LEED AP credentials through a highly structured exam development, application, and registration process. GBCI's second area of expertise is project certification, which involves administering the LEED certification program according to validated international standards of the LEED Rating System.

The scope of our analysis was concerned with the project certification process. GBCI's LEED certification involves three steps: Registration, Application and Certification. The certification process includes two revenue generation points, Registration and Application Review. Organizations seeking LEED certification pay an upfront registration fee as well as a second, larger fee for application review.

Figure 1. LEED Certification Process Overview



Data

Dean DiPietro, a current employee of GBCI, provided the data. The original data included 26,264 records and 24 variables. For a sample of the data use in predictive modeling, please see exhibit A.

Goal of Project & Analysis Used

The goal of our analysis was to evaluate various classification methods in hopes of creating a model capable of helping GBCI predict whether a registrant will file an application for review within 12 months of registration. This is the period of interest for GBCI, given their accounting cycle. An accurate prediction would allow the organization to perform more reliable cash flow forecasting, which will inform hiring and expansion plans. Ultimately, all of our models found that the type of data collected by GBCI was not robust enough to use for prediction, despite the clear visualized class differences for several of the variables (Attached Exhibit D). This implies that other noise within the data prevents the discriminating variables from clearly signaling differences in the outcome classes. Thus, none of the models perform better than 90% of the time, which is the accuracy you achieve by simply predicting that each building is a "No".

Recommendations

Although factors such as lower square footage; more time from registration to the end of an application system; and residing in the Midwest all increase the probability of a "Yes" outcome, we advise GBCI not to use these variables for predicting time between registration and application. Based on the extensive classification models, we are unable to demonstrate that these variables have sufficient predictive ability. The most likely reason for this is that the drivers behind application lead time are not yet reflected in the data.

Our recommendation to GBCI is to modify the registration process to include two questions, whose responses might serve as more discriminating predictive variables.
1. What is the completion status of your project? (Idea, Plans, Construction in Progress, Completed)
2. When do you expect to submit the completed application forms? (0-3, 6-12, 12-24, 25+ months)

Technical Summary

Data Preparation

Rows containing extreme outliers in the "Gross Square Footage" and "Month_Diff_Reg_App" variables were removed and data trimmed at the recommendation of Dean. Each record corresponded to a unique identifier, Project ID. Additional variables captured location information (City, State, Country), LEED certification type, dates (Registration, Next Key Event, Certification), building size and project, owner, owner occupant and organization type. For the purpose of our analysis we created the following variables: "Completed in 12 months" (our dependent categorical variable), Region (U.S. states were binned into five regions, separate from one International bin), "Proj_Commercial" and "Proj_Military", "Square footage binned", "Month_Diff_Reg_App" (Month difference between registration and application) and "Month_Reg_Before_Close" (Months between registration and close of the certification version applied under) (Attached Exhibit A).

Visualizations

In order to determine the most likely discriminating variables, we developed various visualizations of our data. The continuous variables, "Gross Sq Ft", "Log(SqFt)", and "Months_Diff_Reg_Close" were graphed with a box plot, and the others were developed into bar charts (Attached Exhibit B). Additionally, we created pivot tables to recognize and fathom visible distinctions between variables. The variables that appeared to be most useful based on this exploration included: GrossSqFt, Log(SqFt), Months_Diff_Reg_Close, Proj_Commercial, Proj_MilitaryBase, and Region_Midwest. The majority of the attempted models were built on these discriminating variables.

Modeling

Besides basic visualization techniques and pivot tables, we explored Naïve-Bayes, Logistic Regression, Classification and Regression Trees and *k*-Nearest Neighbor for the purpose of our analysis. Discriminating variables discovered in the visualization stage were used to determine the most promising variables which were then incorporated in the various modeling techniques. See the model outputs attached as exhibits for further detail (Attached Exhibit C).

As the initial models did not produce error rates that improved upon the naïve case, we attempted to re-run each model using oversampled data in which the training set and validation sets were manipulated to reflect 40% classification rate for the "Yes" class. As discussed below, the oversampling was able to improve the classification of Actual "Yes" records, but at the expense of a dramatically higher misclassification rate for "No"s. In no case did the oversampled model cause an improvement in the overall error rate.

k-Nearest Neighbor

k-Nearest Neighbor classifier was run several times using different k values and variable sets. However, none of these combinations produced prediction models with reasonable accuracy. All but two variables in our data set were categorical which likely added to the complexity. The Test Data % Error was 9.41% for regularly partitioned data and 44.29% for data that was oversampled to include a 50:50 split of "Yes" and "No" data points.

Classification Trees

Given the large number of potential dummy variables, we were looking for a modeling process that might clue us in to the most important dummies in a category, or the appropriate bins to use for continuous variables. In the initial classification tree models, the software (both XLMiner and Spotfire Miner) returned, as a tree, a single box with the word “No”. In other words, the model with lowest error was simply to predict the naïve case. By eliminating the forced pruning of the tree, we were able to achieve a branching output, but without any meaningful “Yes” nodes (those created by the rules were at level 7 within the tree), and only had a handful of records in each end node. Of note was that the variable “Months_Diff_Reg_Close” was the variable a the top of the “unpruned” tree in each iteration, supporting the initial impressions that it would be an important variable in distinguishing between “Yes” records and “No” records. It is not a clear enough signal, however, to create a practical tree that can be used by GBCI to classify new registrants. The same outcome of the test data was found with the oversampled data.

NAÏVE BAYES

Additionally, a Naïve Bayes classifier was applied to the data set in question. In this model, variables corresponding to confidentiality, regional location, owner type, occupant type, and square footage were utilized. Square footage values were binned, as the model does not allow for continuous variables; 10 separate, evenly spaced bins were created – 400,000 square feet separating one bin from another.

The data reflected projects completed within a twelve month span and while the naïve rate was small (9.058%) the model could not predict one related observation. Each of the 837 observations within the success class “Yes” was misclassified. The Conditional Probability output was used to remove variables. As noted from the visualizations, region data was transformed into dummies and Region_Midwest in specific was utilized. The same was done for commercial properties among the ‘Owner Type’ variable. ‘Owner/Occupant Type’ variable, was removed as it was similar in behavior as the ‘Owner Type’ variable. Square footage was revisited, and re-binned into 4 bins instead of the original 10, and they were allowed to be spaced at different intervals. Unfortunately, the modifications did not alter the results of the model. Further cut-off values were altered; after all, misclassification costs were indeed higher should a project be predicted as ‘successful’ should it not be at all. As before, the result remained.

In a final attempt, it was hypothesized that oversampling may be necessary, whereby 50% of the observations would be rated a ‘success’, as opposed to the approximately 9% in the true data set. Once the model was run on the oversampled data, it would be applied to the original observations. This attempt resulted in an even higher model error than the original – 37.35% compared to 9.34%.

LOGISTIC REGRESSION

Further, logistic regression classifier was leveraged to generate accurate classifications. The original model yielded similar results as other models generated (Attached Exhibit D). Modifying the data to condense the “project type” categories from several to two- namely Proj_Commercial and Proj_MilitaryBase also only resulted in minimal improvement of our model (Attached Exhibit C). In other words, our model classified only a few records of the success class in the test data accurately. Using guidance from our visualizations, the model was re-run with Log(SqFt) (Attached Exhibit C). Though this led to a few more records being accurately classified, this was far from a reasonable and acceptable solution (Attached Exhibit C). As the idea was to accurately classify both the “Yes” and “No” class, this model was not pursued. As a last attempt, oversampled data was utilized to re-run this classifier. %Error for the success class drastically improved to 2.77%. However, the %Overall Error dropped from the original 9.66% to 77.28% (Attached Exhibit C). Again, the accuracy of classifying the success class came with decreasing the predictive accuracy of the whole model.

It was clear that the variables provided did not adequately capture the behavior of the observations provided (Attached Exhibit D). We hypothesized that additional data would be necessary – perhaps capturing the time at which each project completed a physical stage of construction, as opposed to checking off a registration box, which had little to do with project completion at all.

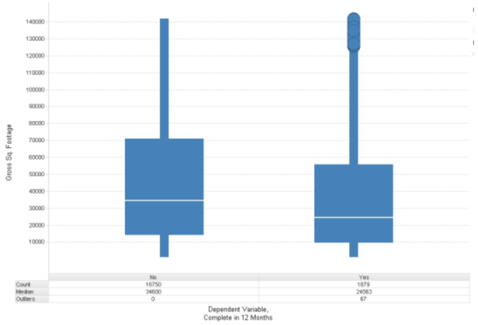
Exhibits

Exhibit A: Sample Data

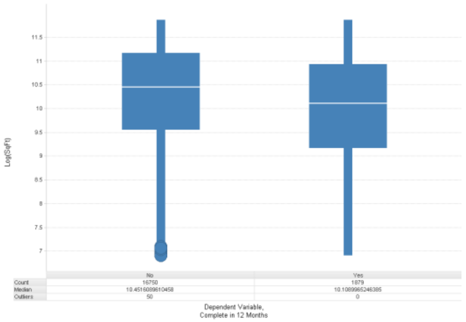
Complete in 12 Months	Region	Proj_ Commercial	Proj_ MilitaryBase	ProjType	GrossSqFt	Owner Type	Month_Reg Before_Close
No	West	YES	NO	Commercial Office	142604	Profit Org.	0.03
Yes	South	YES	NO	Commercial Office	160000	Profit Org.	0.03
Yes	International	NO	NO	Campus	230286	Profit Org.	10.77
No	Midwest	YES	NO	Commercial Office	20000	State Government	17.35
No	South	NO	YES	Military Base	37950	Federal Government	0.83
No	West	NO	YES	Military Base	9364	Federal Government	0.87

Exhibit B: Visualizations

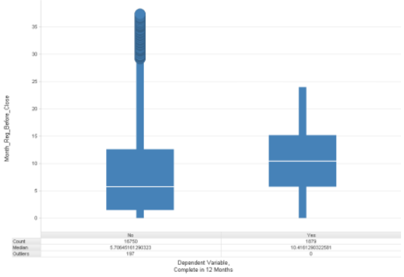
SqFt



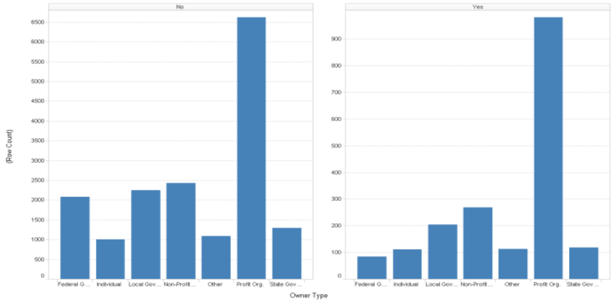
Log(SqFt)



TimeUntilClose



Owner Type



Region

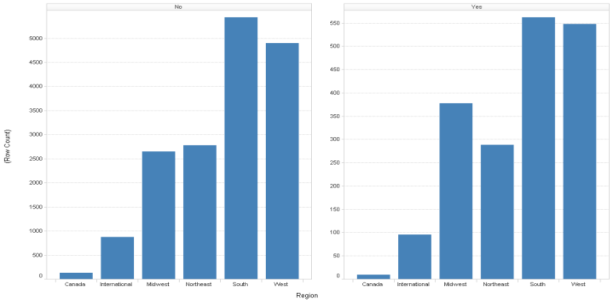
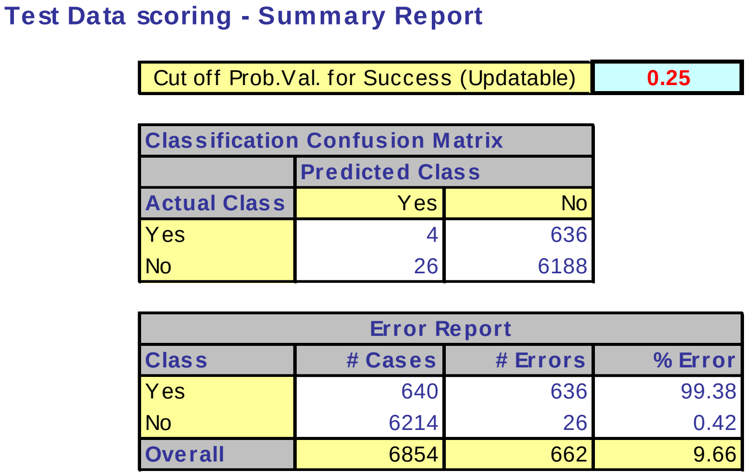


EXHIBIT C: LOGISTIC REGRESSION CLASSIFIER OUTPUTS

MODIFIED DATA



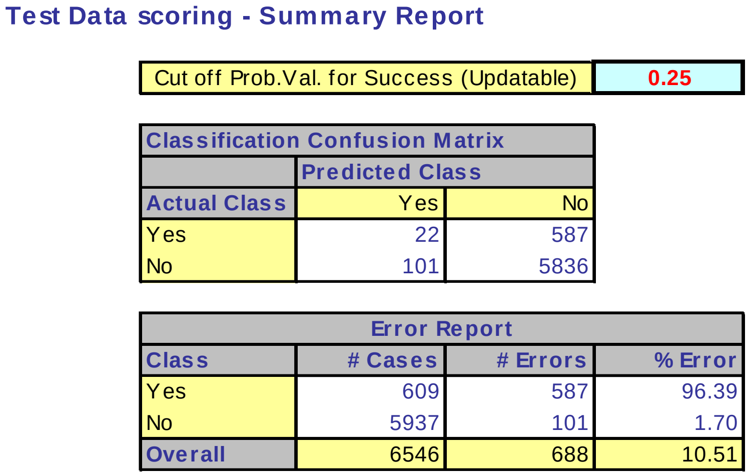
Test Data scoring - Summary Report

Cut off Prob.Val. for Success (Updatable)	0.25

Classification Confusion Matrix		
	Predicted Class	
Actual Class	Yes	No
Yes	4	636
No	26	6188

Error Report			
Class	# Cases	# Errors	% Error
Yes	640	636	99.38
No	6214	26	0.42
Overall	6854	662	9.66

LOG(SQFT)



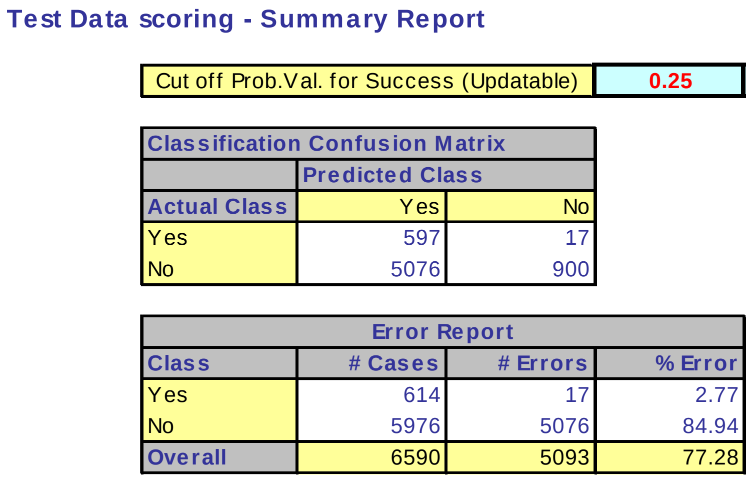
Test Data scoring - Summary Report

Cut off Prob.Val. for Success (Updatable)	0.25

Classification Confusion Matrix		
	Predicted Class	
Actual Class	Yes	No
Yes	22	587
No	101	5836

Error Report			
Class	# Cases	# Errors	% Error
Yes	609	587	96.39
No	5937	101	1.70
Overall	6546	688	10.51

OVERSAMPLED DATA



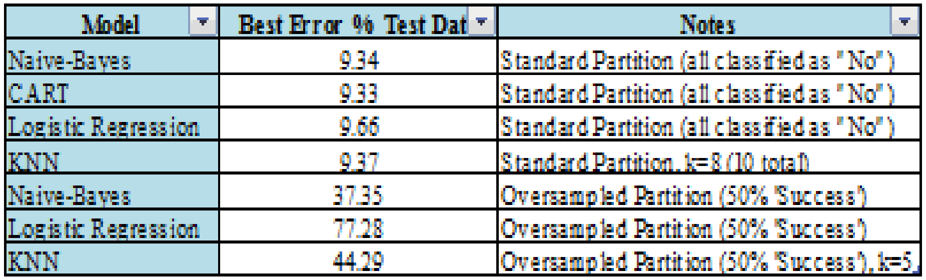
Test Data scoring - Summary Report

Cut off Prob.Val. for Success (Updatable)	0.25

Classification Confusion Matrix		
	Predicted Class	
Actual Class	Yes	No
Yes	597	17
No	5076	900

Error Report			
Class	# Cases	# Errors	% Error
Yes	614	17	2.77
No	5976	5076	84.94
Overall	6590	5093	77.28

EXHIBIT D: PERFORMANCE ACROSS CLASSIFIERS



Model	Best Error % Test Dat	Notes
Naive-Bayes	9.34	Standard Partition (all classified as "No")
CART	9.33	Standard Partition (all classified as "No")
Logistic Regression	9.66	Standard Partition (all classified as "No")
KNN	9.37	Standard Partition, k=8 (10 total)
Naive-Bayes	37.35	Oversampled Partition (50% "Success")
Logistic Regression	77.28	Oversampled Partition (50% "Success")
KNN	44.29	Oversampled Partition (50% "Success"), k=5